National College of Ireland

Configuration Manual

MSc Research Project
MSc Cyber Security

Sidhant Prakash Patil
Student ID: x23128577

School of Computing
National College of Ireland

Supervisor: Prof Liam Mccabe

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sidhant Prakash Patil

Student ID: X23128577

Programme: MSc cyber Security **Year:** 2024

Module: MSc Research Project

Lecturer: Prof Liam Mccabe

Submission Due Date: 12/12/2024

Project Title: Configuration Manual

Word Count: 917…… **Page Count: 19**……..……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sidhant Patil……

Date: 12/12/2024……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sidhant Patil
Student ID: x23128577

1 Introduction

The configuration manual provides detailed instructions of configuration, setup, and operation of the Blockchain based EHR systems. Ethereum blockchain is used with smart contracts written in solidity, IPFS for decentralized storage and React.js for creating a simple scalable, interactive frontend.

2 System configuration

Hardware of the system.

- Processor: Ryzen 9 6900H
- Operating system: Windows 11 Home
- Storage: 1 TB Gen 4 SSD
- RAM: 16 GB 6400Mhz

Versions of Software used.

- **Truffle Suite**: Truffle v5.11.5 (core: 5.11.5)
- **Web3**: Web3.js v1.10.0
- **Ganache**: UI Ganache v2.7.1
- **MetaMask**: 12.6.2 (browser extension)
- **IPFS Desktop**: CLI V 0.31.0
- **Visual Studio Code** : Version: 1.95.3 (user setup)
- **React.js**: version 18.2.0
- **Solidity**: Compiler version 0.8.19
- **Node.js**: Version 20.17.0
- **NPM (Node Package Manager)**: version 10.8.3

3 Setup and configure development environment

This section provides instructions to set up the environment and configuration of the system components.

3.1 Install Node.js and NPM from https://nodejs.org/en/download/package-manager and verify the installation by running node -v or npm -v in the command prompt.

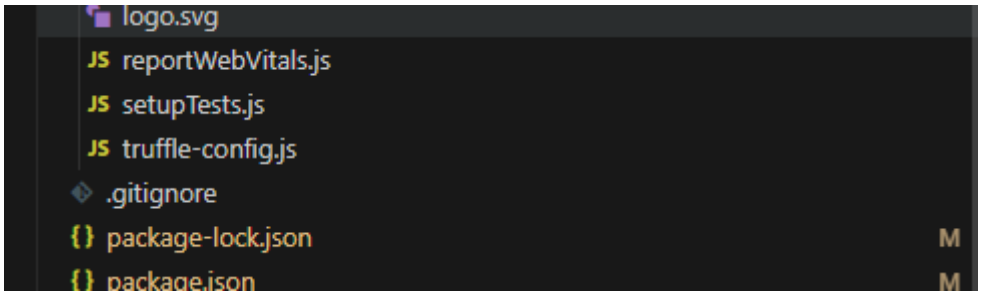
3.2 Setup git to track and upload the files to save changes made to the project. Install and verify the git using git –version command.

```
PS C:\Users\siddh> git --version
git version 2.47.0.windows.1
PS C:\Users\siddh>
```

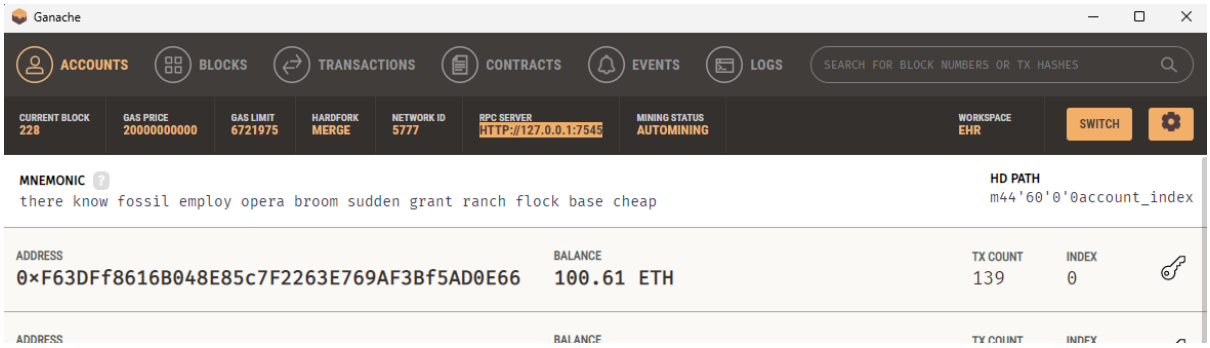
3.3 Install Truffle globally to compile and deploy the smart contracts by using the npm install -g truffle and verify the installation using truffle version command

```
PS C:\Users\siddh> truffle version
Truffle v5.11.5 (core: 5.11.5)
Ganache v7.9.1
Solidity v0.5.16 (solc-js)
Node v20.17.0
Web3.js v1.10.0
PS C:\Users\siddh>
```

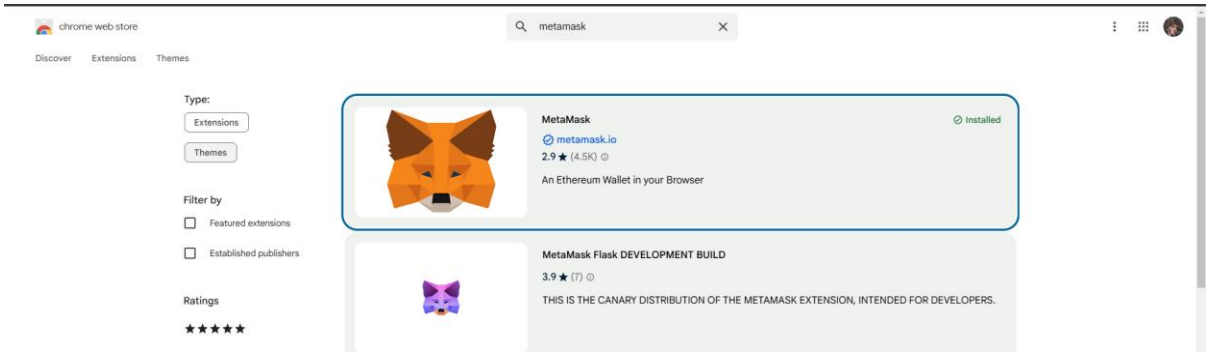
3.4 Download and install Ganache from the official website. launch ganache and add truffle project by selecting the truffle.config file created after installing truffle.



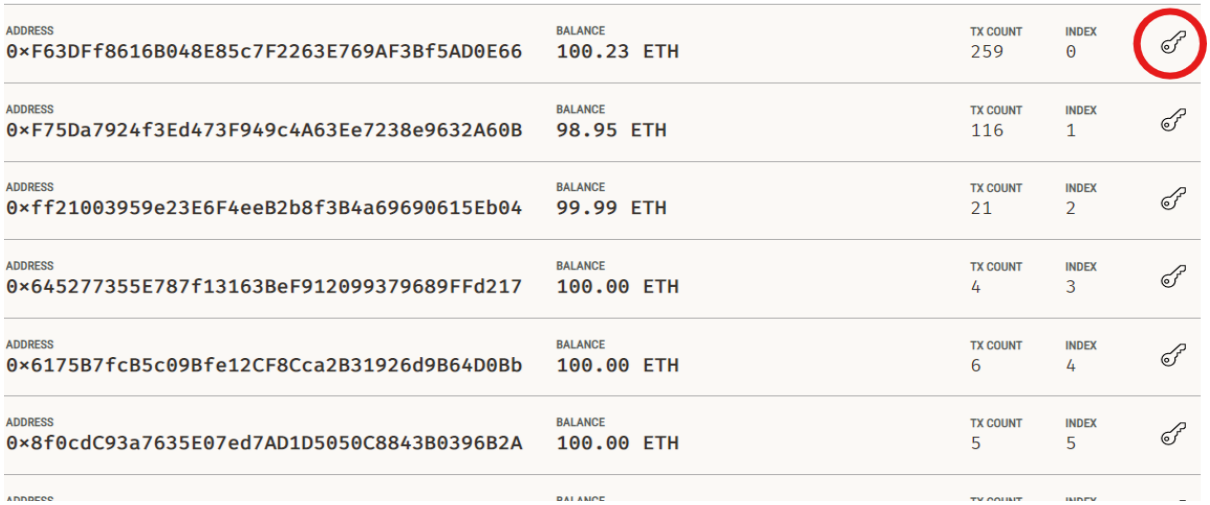
3.5 Configure the RPC server URL to HTTP://127.0.0.1:7545 and network ID to 1337 from 5777 as 1337 is a default port used by ganache even if it shows 5777.



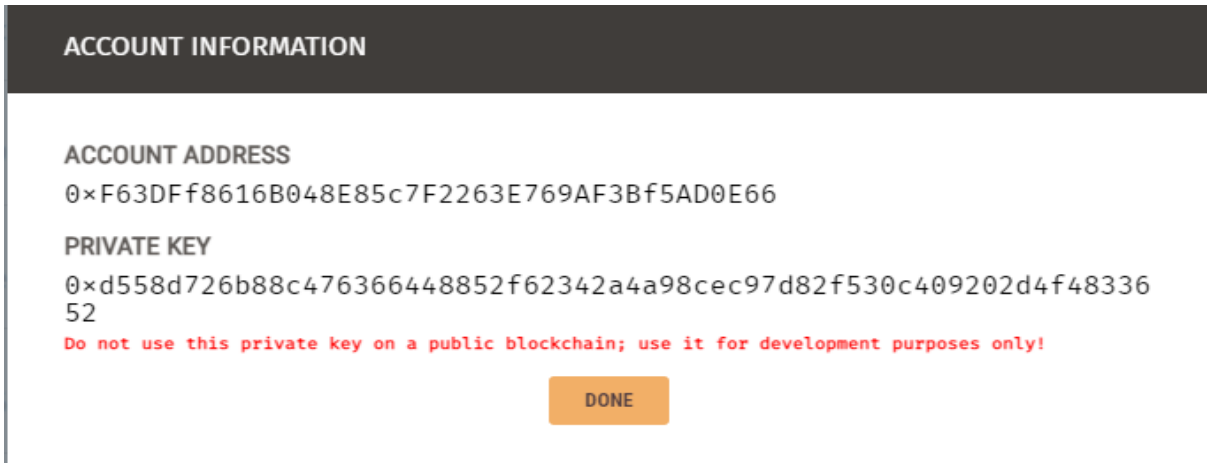
3.6 Add and configure MetaMask extension on preferred browser.



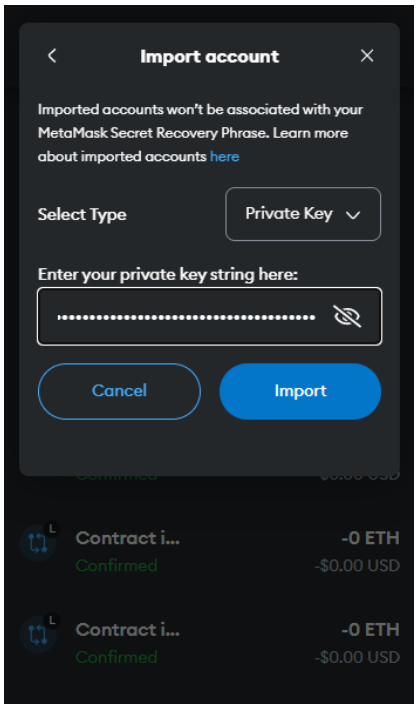
3.7 Import 3 prefunded accounts form ganache for testing by copying the private key provided in ganache.

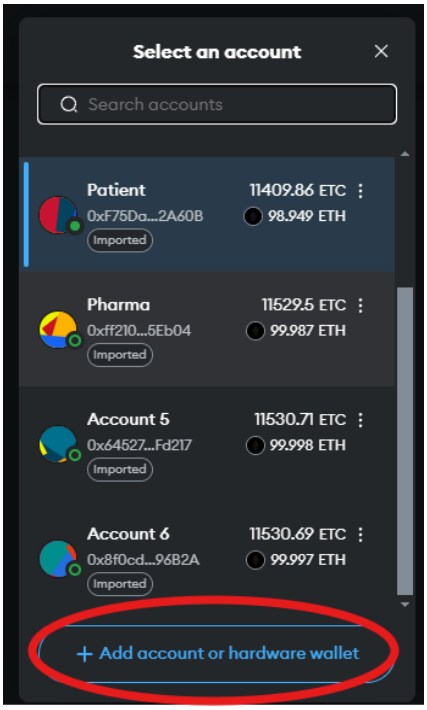


3.8 Copy Private key

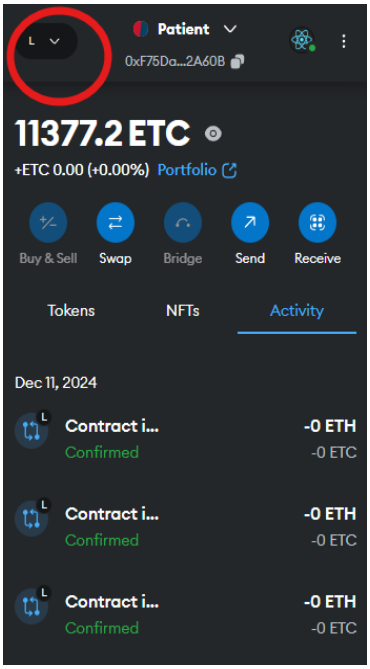


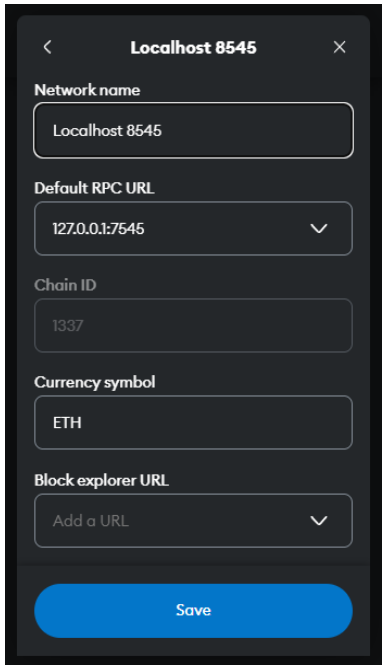
3.9 Import accounts to MetaMask





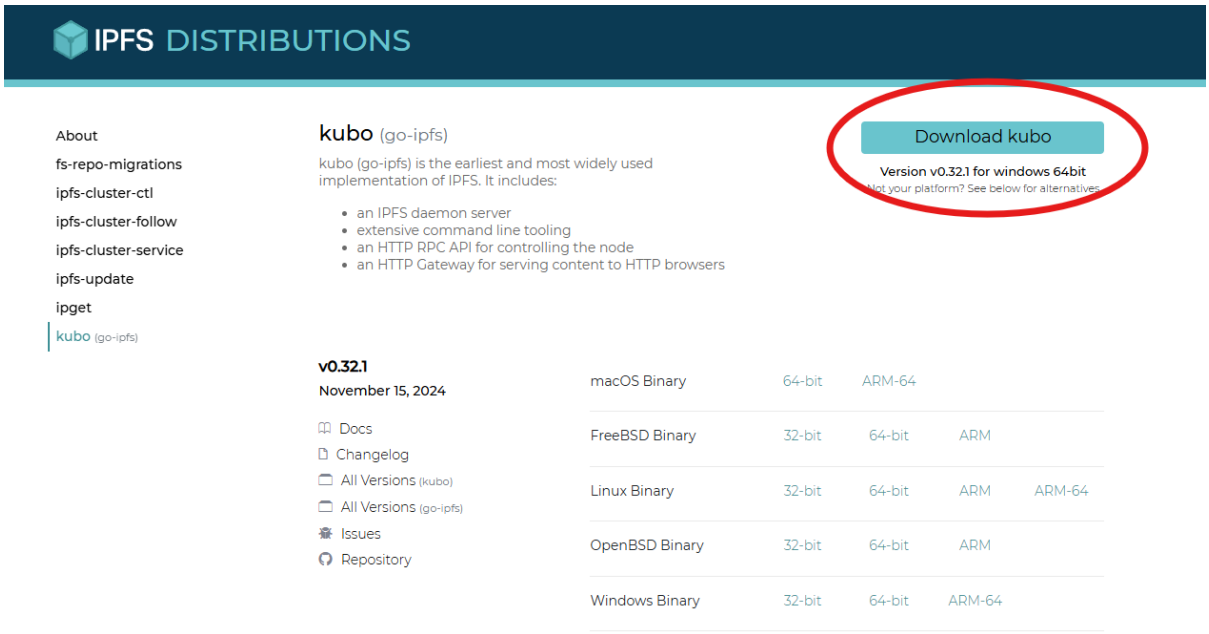
3.10 Add a custom network with network name: Ganache Network or the name suggested by the MetaMask to avoid any conflicts within the networks, RPC URL: HTTP://127.0.0.1:7545 and Chain ID: 1337 (note: as you can see the chain id is 1337 by default for local ganache network even if it shows 5777 in the ganache application)



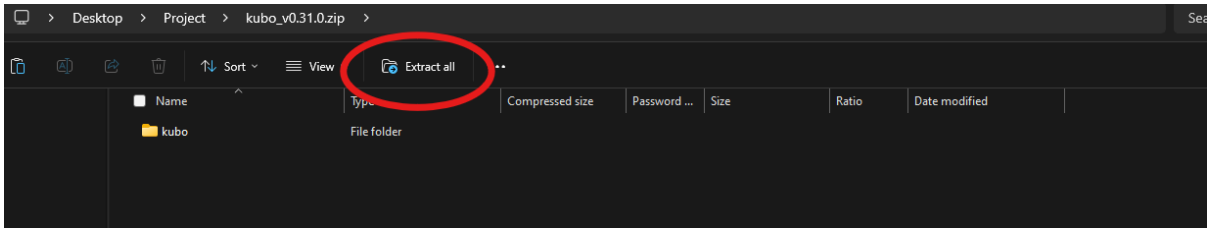


3.11 Setup and install IPFS Kubo as guided in the provided link https://docs.ipfs.tech/install/command-line/ for storing the records. follow below steps for windows.

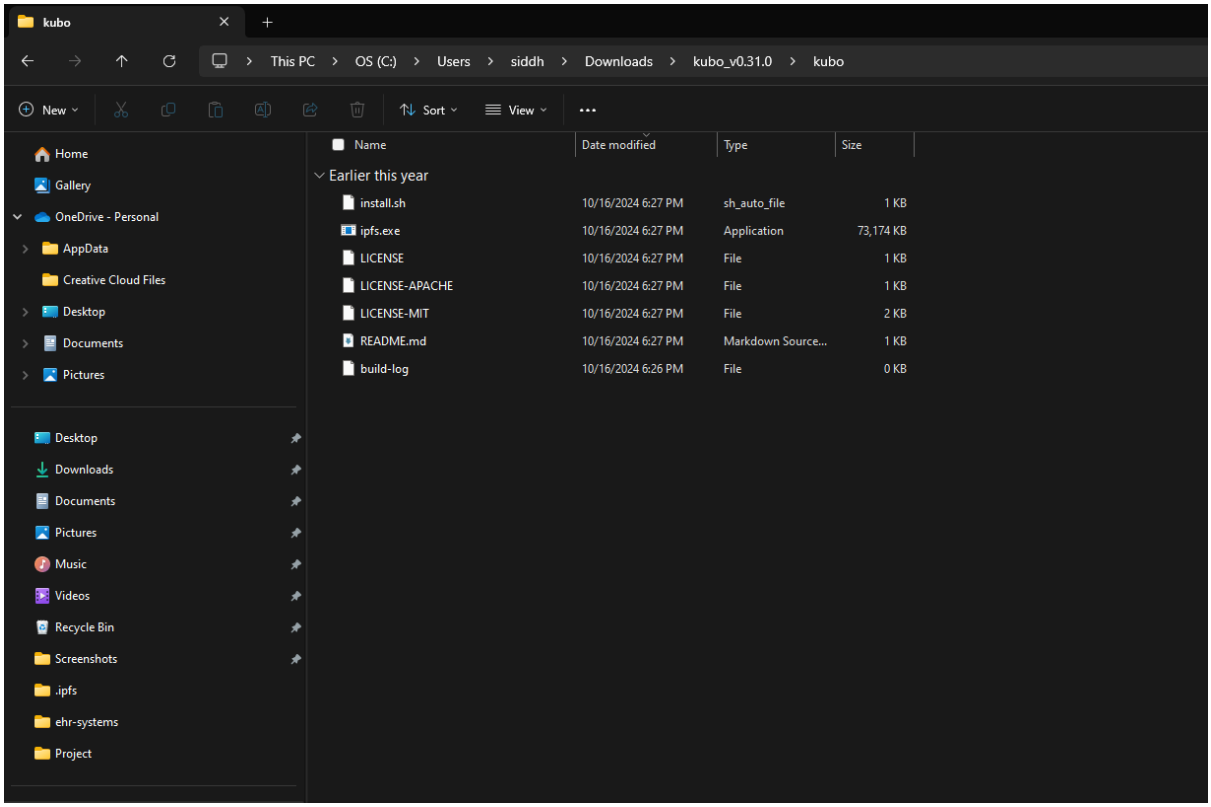
3.12 Download the binary for windows from https://dist.ipfs.tech/#kubo or use wget https://dist.ipfs.tech/kubo/v0.32.1/kubo_v0.32.1_windows-amd64.zip -Outfile kubo_v0.32.1.zip command



3.13 Extract the zip file at preferred location by extractor like 7zip or by using command Expand-Archive -Path kubo_v0.32.1.zip -DestinationPath ~\Apps\kubo_v0.32.1



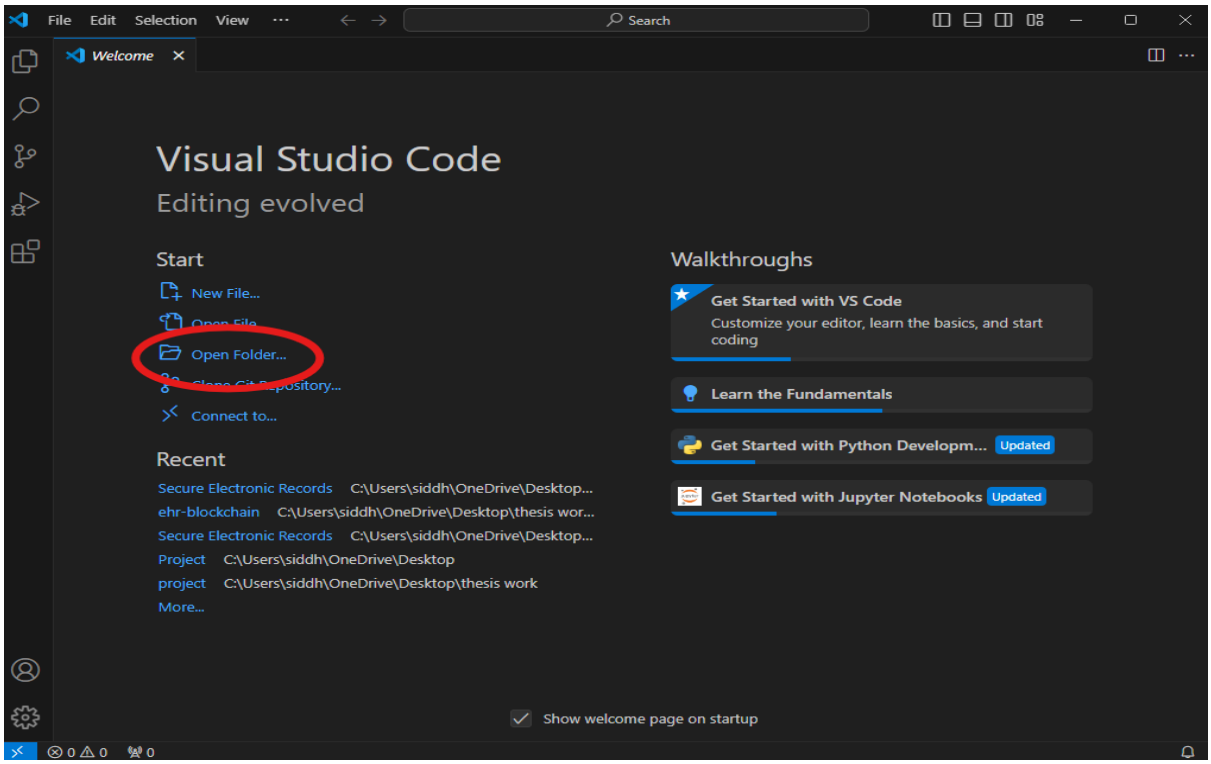
3.14 Redirect to the folder where the zip is extracted: cd /destination to the folder



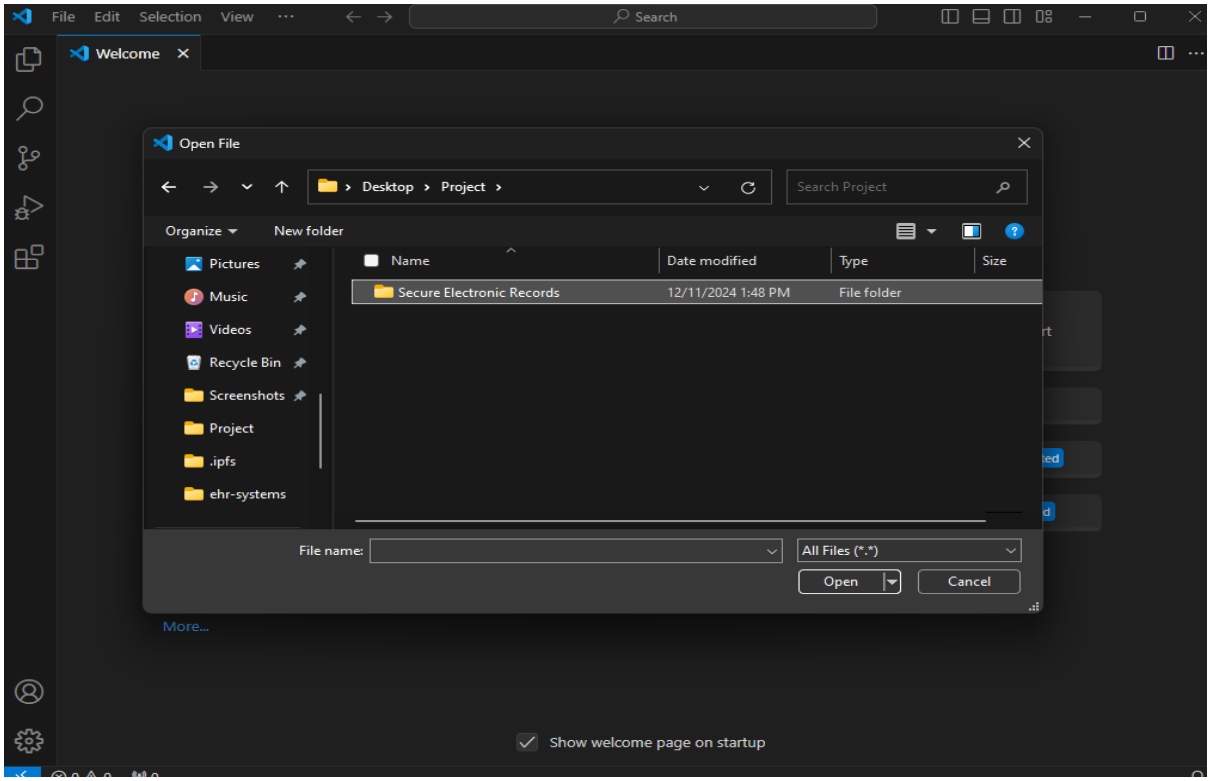
3.15 Check the IPFS.exe and verify version

3.16 Install or Update Visual Studio Code from https://code.visualstudio.com/download

3.17 Open the folder with project files or by cloning the repository using ‘git clone repo url’ and ‘cd Project folder’



3.18 Select the project directory and click open



3.19 Now in VS Code open terminal and install all the dependencies by using npm install.

```
PS C:\Users\siddh\OneDrive\Desktop\thesis work\EHRProject\ehr-blockchain> npm install
npm warn deprecated @npmcli/move-file@2.0.1: This functionality has been moved to @npmcli/fs
npm warn deprecated abab@2.0.6: Use your platform's native atob() and btoa() methods instead
npm warn deprecated @babel/plugin-proposal-unicode-property-regex@7.18.6: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-unicode-property-regex instead.
npm warn deprecated domexception@2.0.1: Use your platform's native DOMException instead
npm warn deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm warn deprecated @babel/plugin-proposal-async-generator-functions@7.20.7: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-async-generator-functions instead.
npm warn deprecated @wessberg/ts-evaluator@0.0.27: this package has been renamed to ts-evaluator. Please install ts-evaluator instead
npm warn deprecated ipfs-core-utils@0.18.1: js-IPFS has been deprecated in favour of Helia - please see https://github.com/ipfs/js-ipfs/issues/4336 for details
npm warn deprecated ipfs-core-types@0.14.1: js-IPFS has been deprecated in favour of Helia - please see https://github.com/ipfs/js-ipfs/issues/4336 for details
npm warn deprecated ipfs-http-client@60.0.1: js-IPFS has been deprecated in favour of Helia - please see https://github.com/ipfs/js-ipfs/issues/4336 for details

added 1186 packages, and audited 1187 packages in 14s
```

3.20 Configure the truffle-config.js as provided.

```
JS truffle-config.js > <unknown> > networks
module.exports = {
  networks: {
    development: {
      host: "127.0.0.1",     // Localhost (default: none)
      port: 7545,            // Standard Ethereum port (default: none)
      network_id: "*",       // Any network (default: none)
    },

  // Configure your compilers
  compilers: {
    solc: {
      version: "0.8.19",      // Fetch exact version from solc-bin (default: truffle's version)
      // docker: true,        // Use "0.5.1" you've installed locally with docker (default: false)
      // settings: {          // See the solidity docs for advice about optimization and evmVersion
      //  optimizer: {
      //    enabled: false,
      //    runs: 200
```

3.21 Compile and deploy the smart contracts Navigate to the source(src) folder ‘cd .\src\’ then ‘truffle compile’ to compile the smart contracts and ‘truffle migrate’ command to deploy the contract on ganache local network.

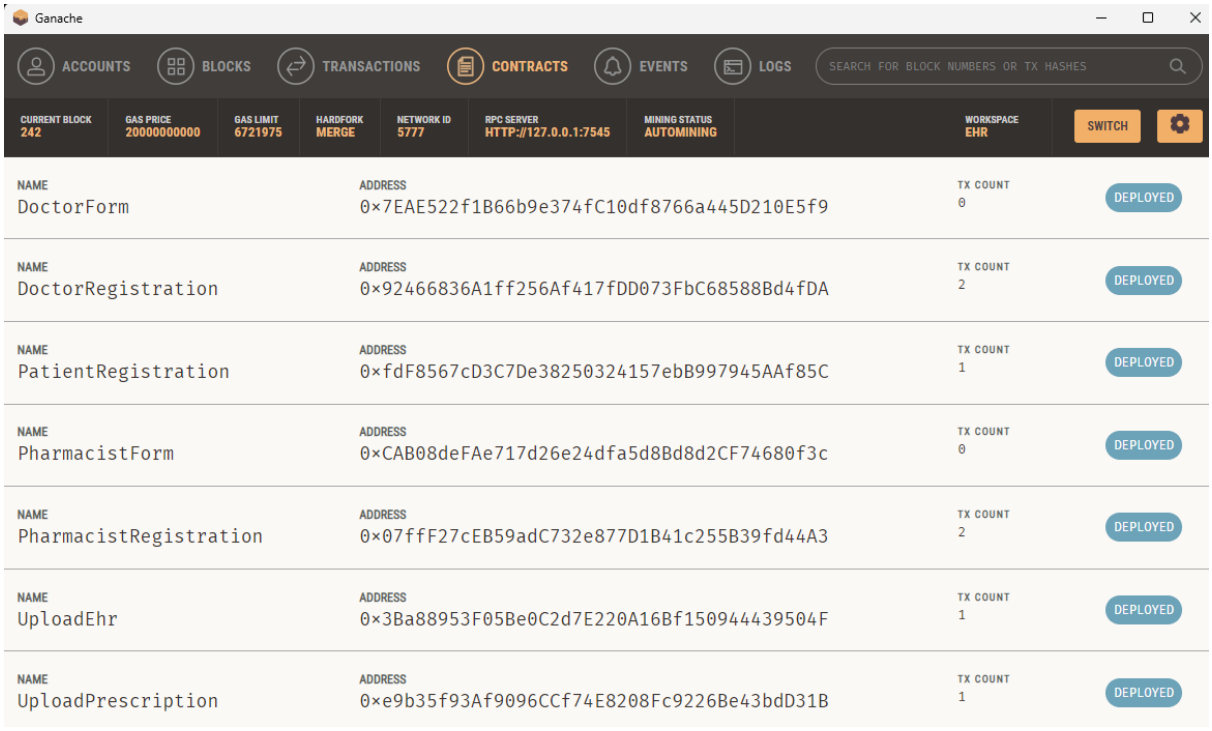
```
PS C:\Users\siddh\OneDrive\Desktop\Project\Secure Electronic Records\Secure Electronic Records\src> truffle compile

Compiling your contracts...
===========================
> Compiling .\contracts\DoctorForm.sol
> Compiling .\contracts\DoctorRegistration.sol
> Compiling .\contracts\PatientRegistration.sol
> Compiling .\contracts\PharmacistForm.sol
> Compiling .\contracts\PharmascistRegistration.sol
> Compiling .\contracts\UploadEhr.sol
> Compiling .\contracts\UploadPrescription.sol
> Artifacts written to C:\Users\siddh\OneDrive\Desktop\Project\Secure Electronic Records\Secure Electronic Records\src\build\contracts
> Compiled successfully using:
   - solc: 0.8.19+commit.7dd6d404.Emscripten.clang
```

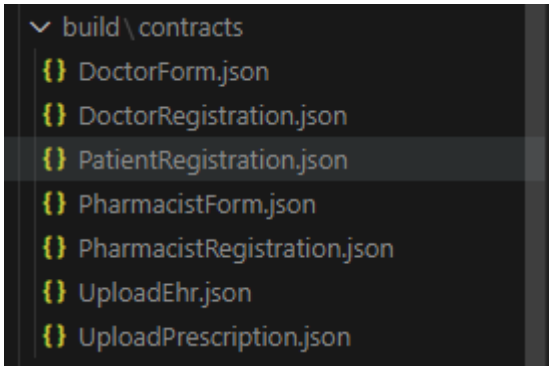
```
PS C:\Users\siddh\OneDrive\Desktop\Project\Secure Electronic Records\Secure Electronic Records\src> truffle migrate

Compiling your contracts...
===========================
> Compiling .\contracts\DoctorForm.sol
> Compiling .\contracts\DoctorRegistration.sol
> Compiling .\contracts\PatientRegistration.sol
> Compiling .\contracts\PharmacistForm.sol
> Compiling .\contracts\PharmascistRegistration.sol
```

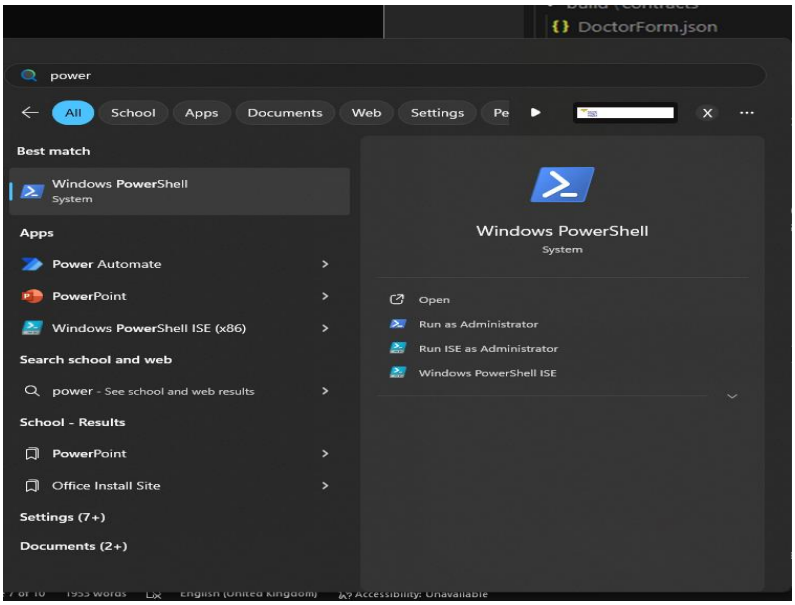
3.22 Verify the deployment of the contracts in ganache.



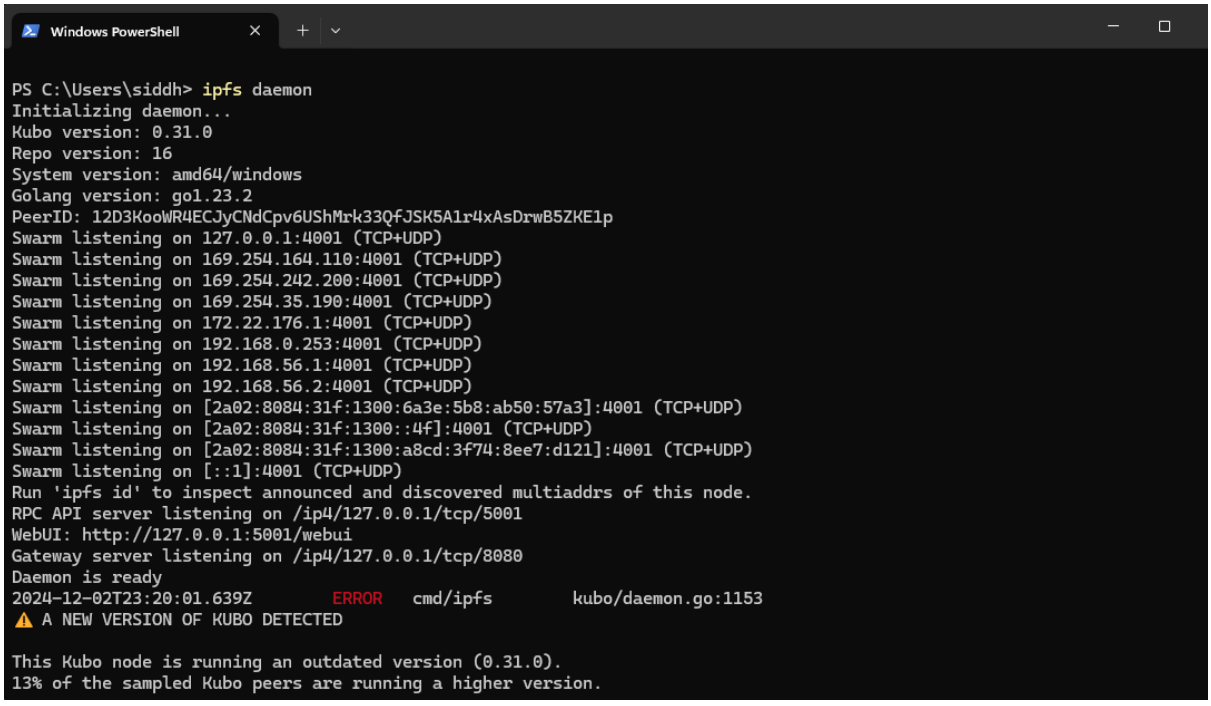
3.23 Verify the abi files in build/contracts folder



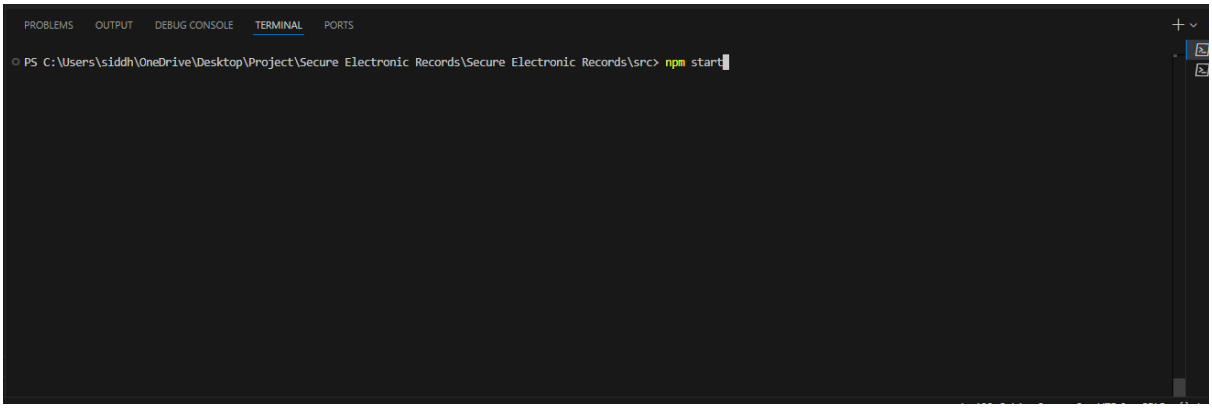
3.24 Start the IPFS daemon open windows powershell from windows Search window and search powershell or press Win+R and type powershell and enter



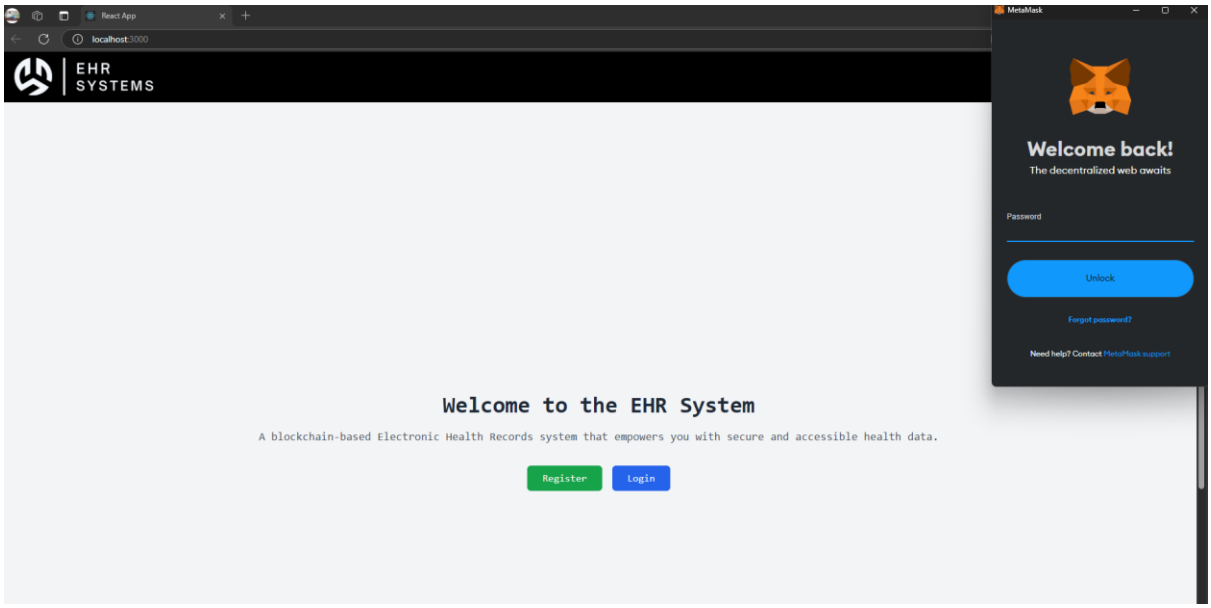
3.25 Enter command ‘ipfs daemon’



3.26 Start the project using ‘npm start’ command in visual studio code.This redirects to the dashboard on the default browser at http://localhost:3000/. MetaMask extension will pop up login and connect MetaMask to the website.



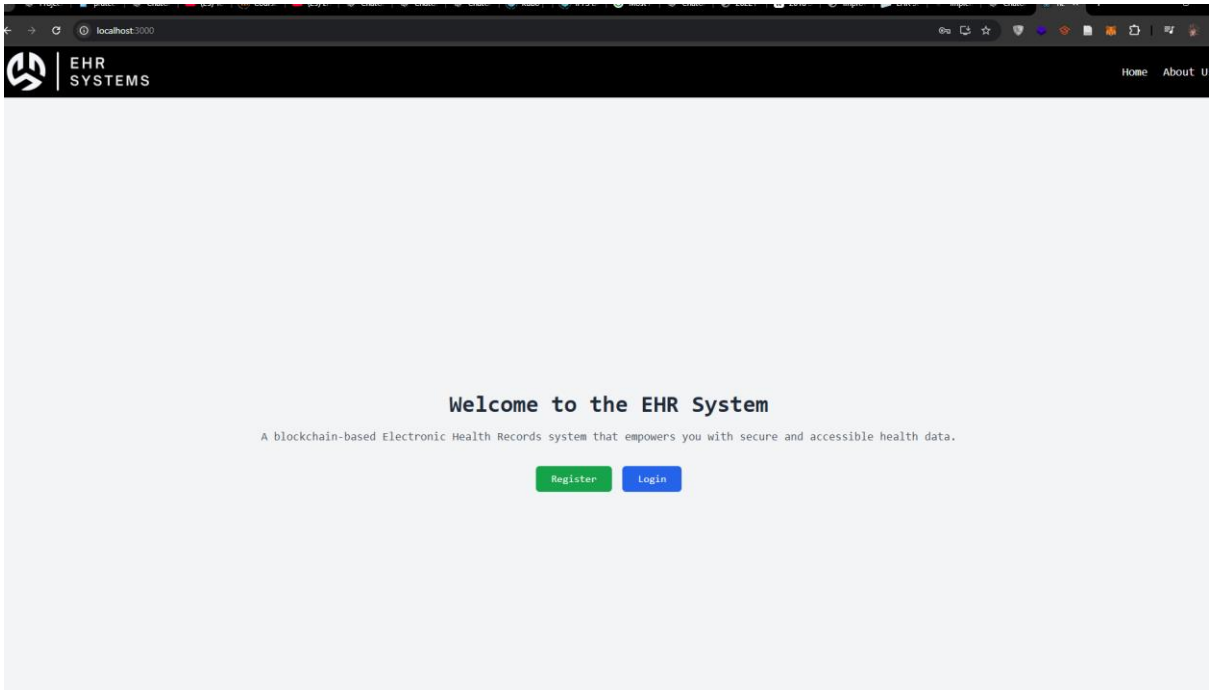
3.27 Application interface



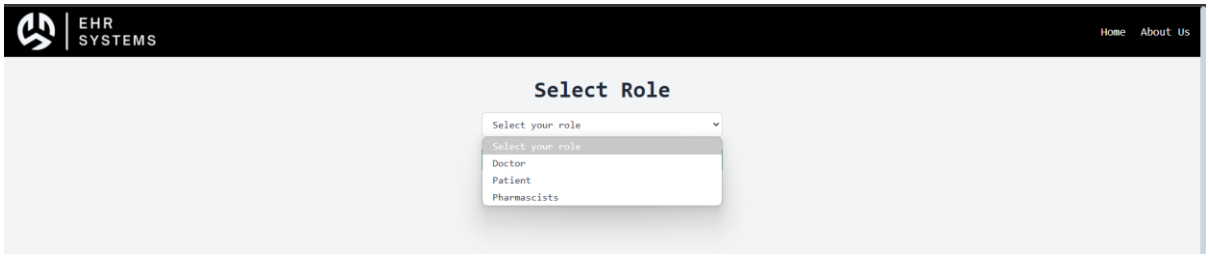
4 Testing the web Application

4.1 Register as a User:

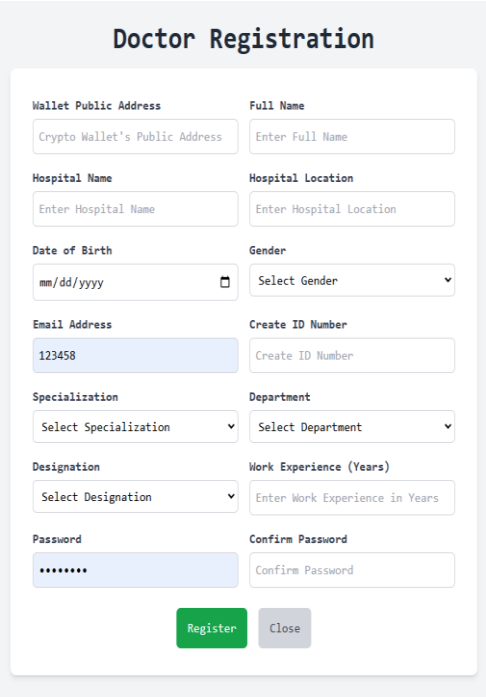
4.1.1 On the home page of the website click on register.

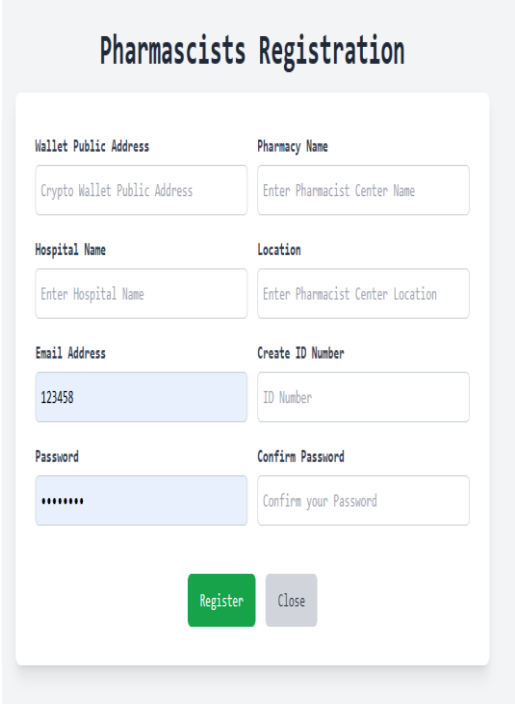


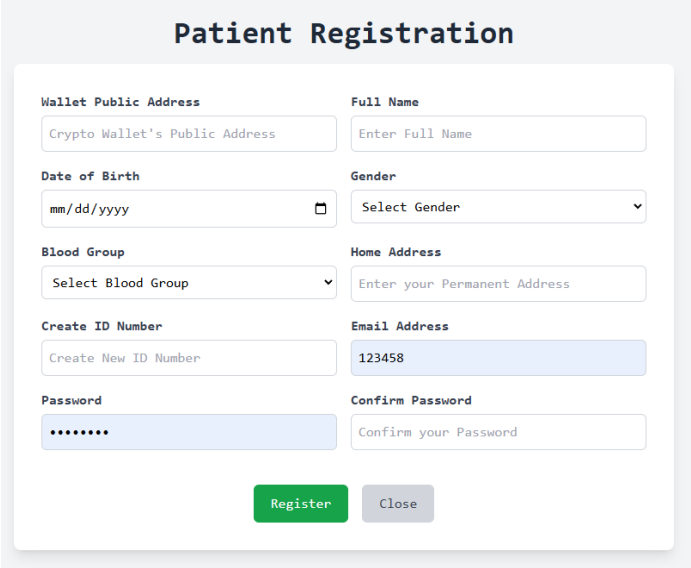
4.1.2 Select a role (Patient, Doctor, Pharmacist).



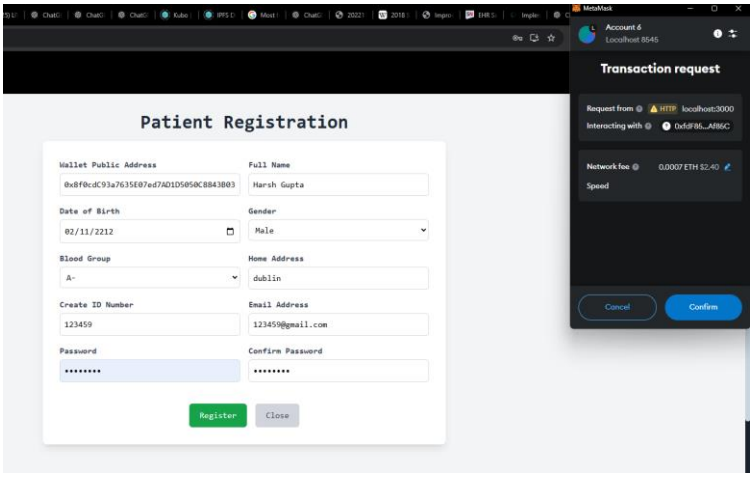
4.1.3 Register the user.





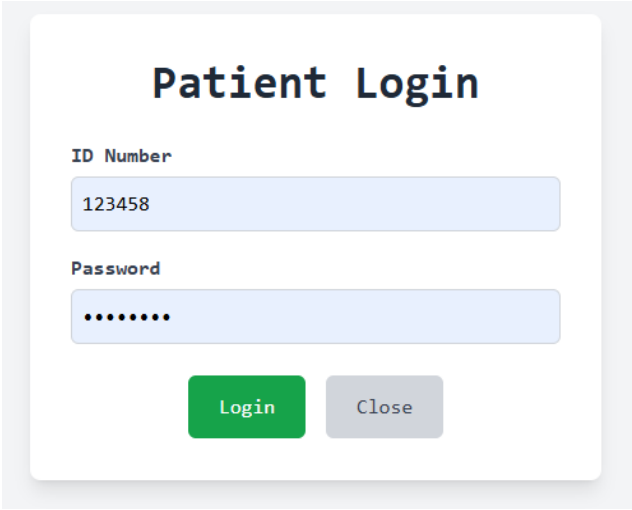


4.1.4 Confirm the MetaMask transaction for registration.

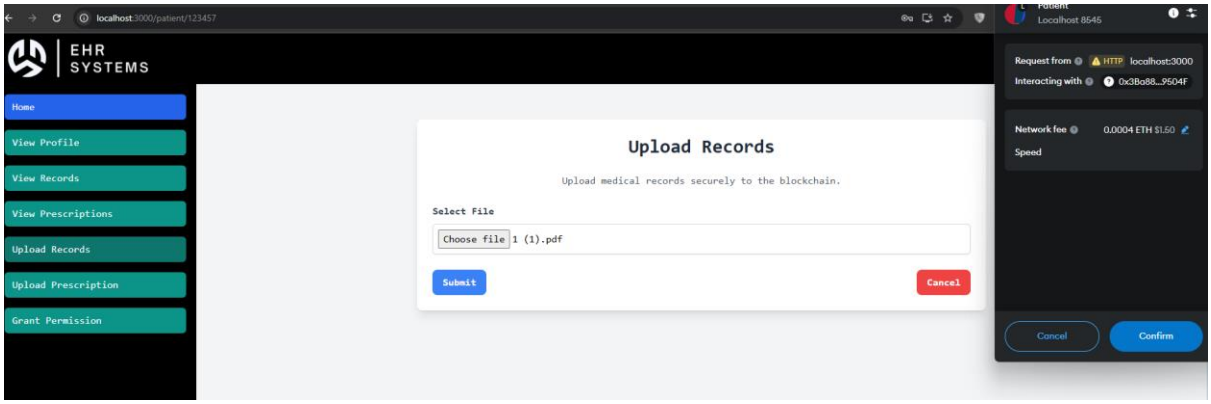


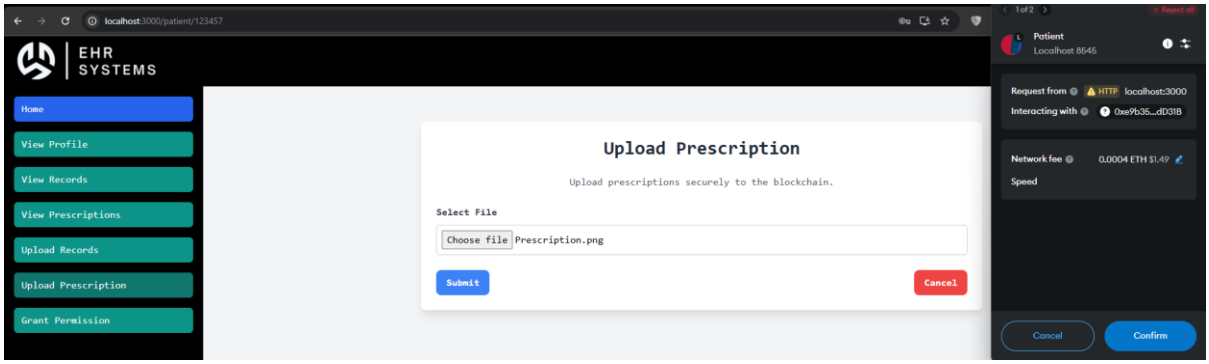
4.2 Upload Medical Records (Patients):

4.2.1 Login to the patient dashboard.

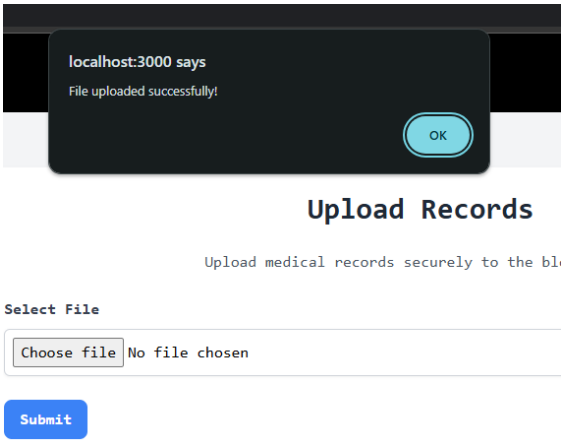


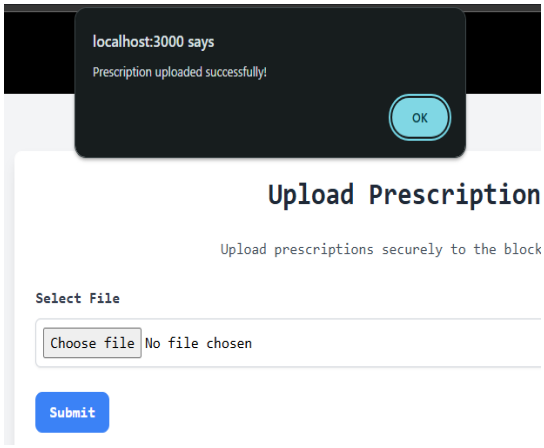
4.2.2 Upload the health records or prescription and confirm the transaction in MetaMask.





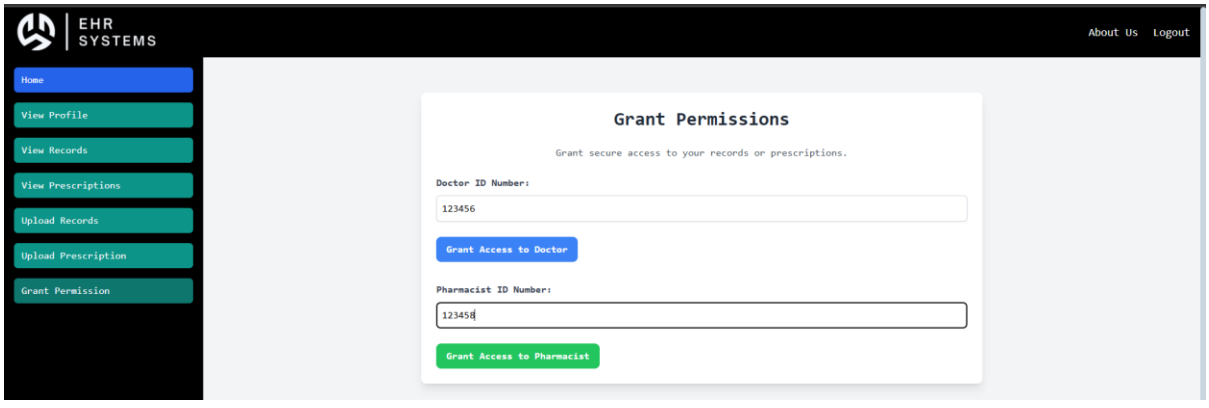
4.2.3 The file selected is encrypted and uploaded to IPFS.



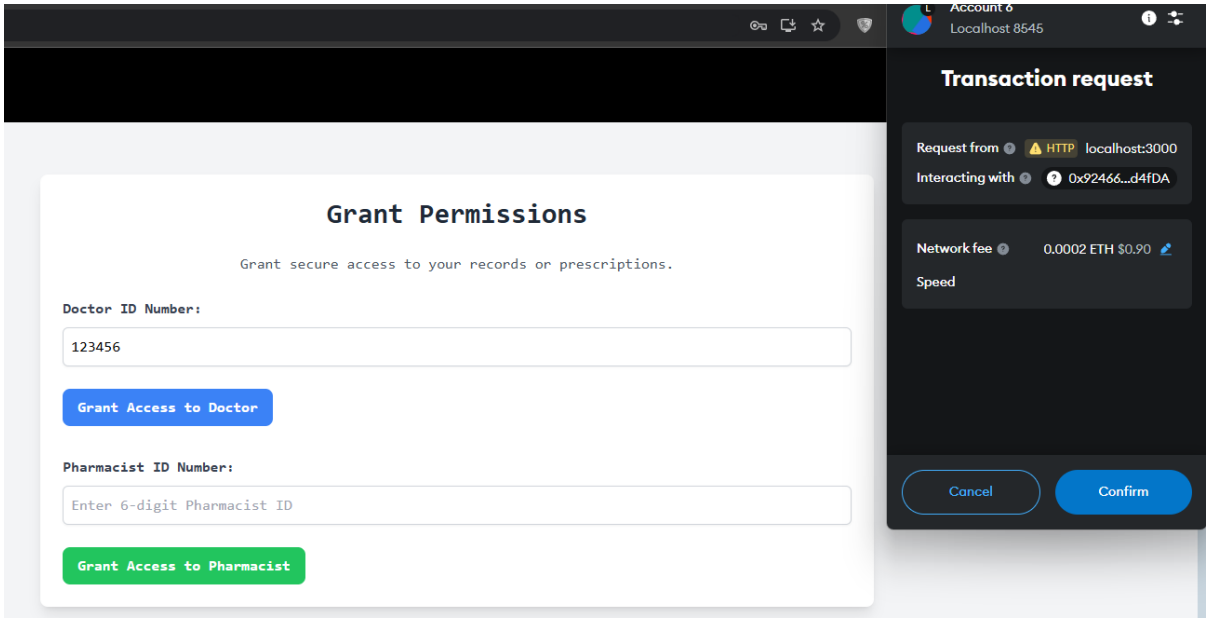


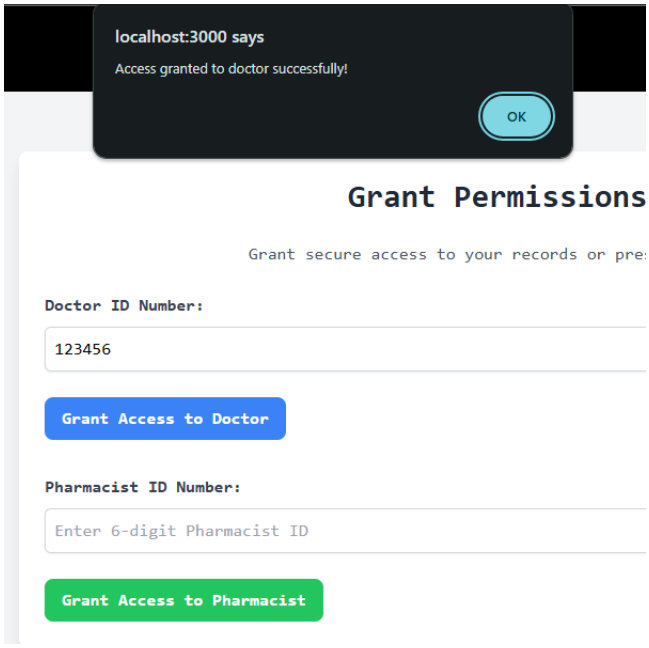
4.3 Grant Access to Doctors:

4.3.1 In the patient dashboard use grant permission tab to provide access to records and prescription uploaded.



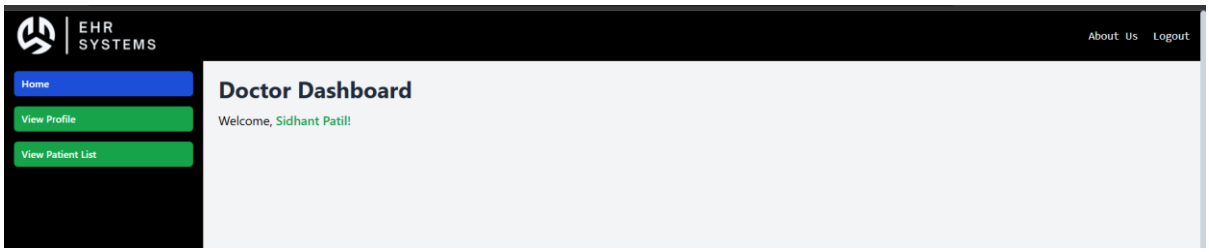
4.3.2 Confirm the transaction in MetaMask.





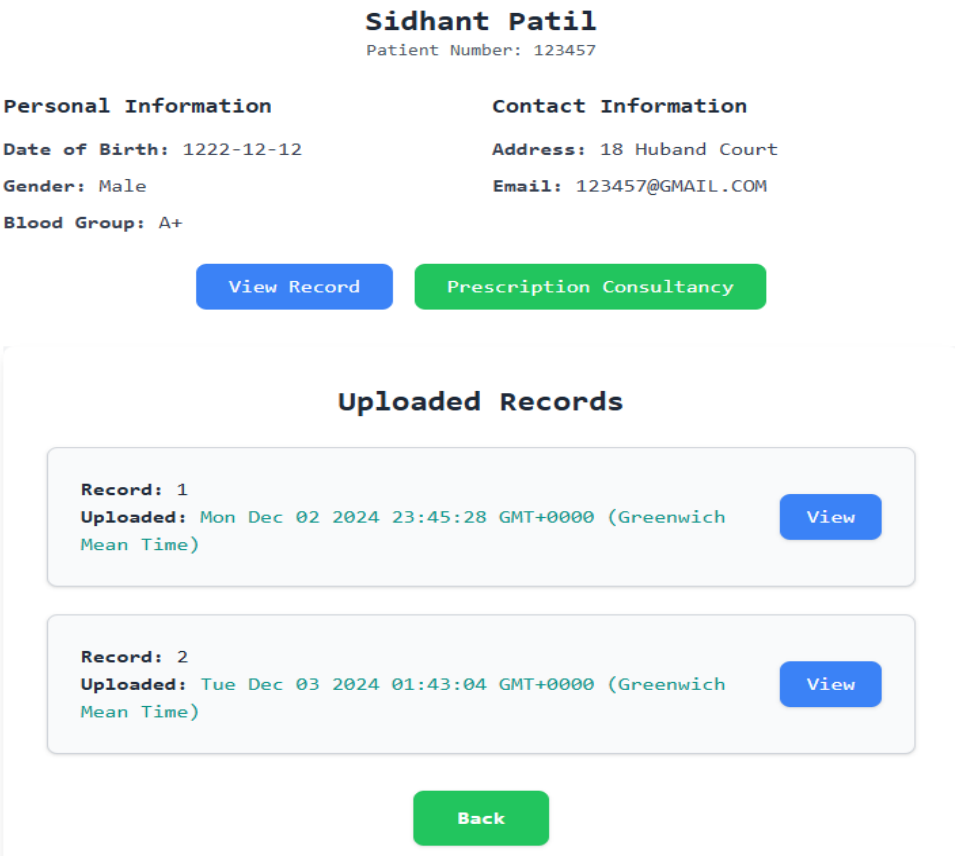
4.4 Access Records (Doctors):

4.4.1 Log in as a doctor.



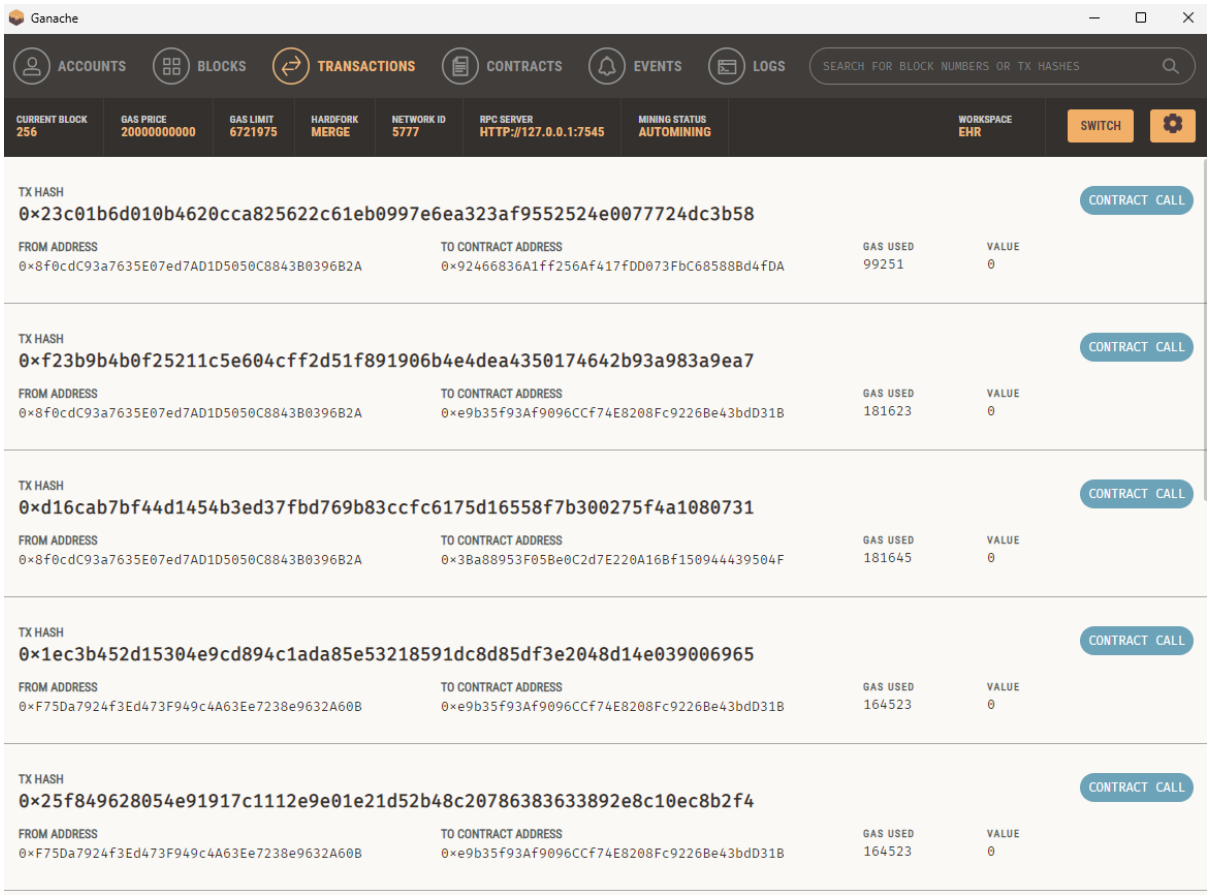
4.4.2 View the list of patients who have granted access.

4.4.3 View records in doctor dashboard and view prescription in pharmacist dashboard retrieves the medical records/prescription to which patient has given access.

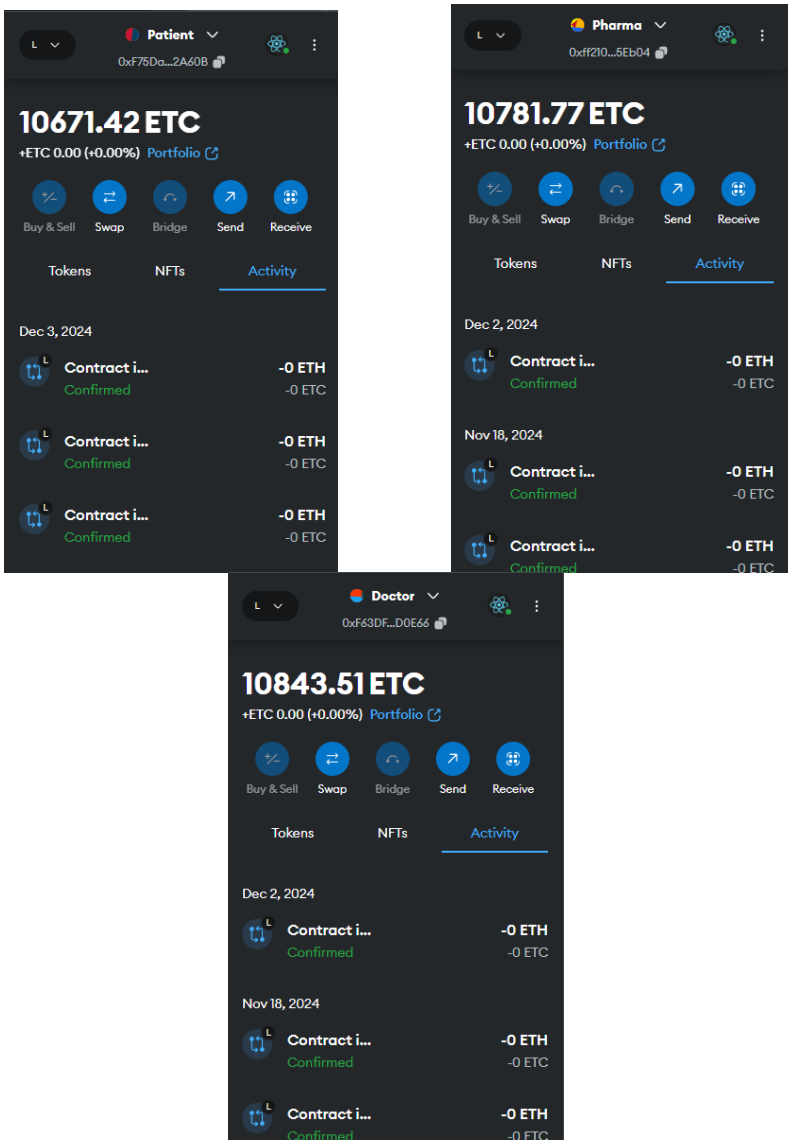


4.5 Monitor Transactions in Ganache and MetaMask:

4.5.1 Observe the transactions being recorded in Ganache.



4.5.2 Check account balances and transaction history in MetaMask.



5 Troubleshooting

Common Issues faced and Solutions

1. **MetaMask and Ganache connectivity**: Verify the MetaMask local network configuration matches the Ganache network settings. Try launching new ganache project with different RPC URL port. To verify the connection make a transaction between the accounts.
2. **Smart Contract related issues**: Check for any syntax errors in code any overshadowing. Reconfigure the truffle-config,js file to match the port, network_id and verify the solidity compiler version. Make sure the correct Abi is called where required. Verify ganache is up and running.
3. **IPFS Connection Issues**: Verify the IPFS gateway address and Port as it may be used by other application.

9. References

1. Truffle: https://archive.trufflesuite.com/
2. Web3.js: https://web3js.readthedocs.io/en/v1.10.0/
3. Ganache: https://archive.trufflesuite.com/ganache/
4. MetaMask: https://metamask.io/

5. IPFS: https://github.com/ipfs/ipfs-desktop
6. Visual Studio Code: https://code.visualstudio.com/
7. React.js: https://react.dev/
8. Solidity: https://soliditylang.org/
9. Node.js: https://nodejs.org/en
10. NPM: https://www.npmjs.com/